

الفصل الثالث: تقنية الأجاكس

أهداف الفصل

١. التعرف علي مفهوم تقنية الأجاكس.
٢. الطريقة العامة لعمل التطبيقات القائمة على تقنية الأجاكس .
٣. مقارنة طريقة عمل نموذج تطبيقات الأجاكس مع نموذج تطبيقات الويب العادية .
٤. التزامن في نموذج الأجاكس و النموذج العادي في تطبيقات الويب .
٥. الأدوات المستخدمة في تطبيقات الويب القائمة على تقنية الأجاكس.
٦. ملخص الفصل

1.3 تقنية الأجاكس (AJAX Technology)

هي إختصار للمصطلح (Asynchronous JavaScript And XML) وهي تقنية أو طريقة بواسطتها يمكن تطوير تطبيقات الويب بحيث تصبح أكثر تفاعلية وإستجابة للمستخدم ،حيث يتم التقليل من حالات العودة للمخدم وإعادة تحميل الصفحات من جديد . في الواقع هذه التقنية ليست بتقنية قائمة بحد ذاتها أو مفهوم جديد في مجال برمجة الويب بل أنها مجموعة من التقنيات الموجودة أصلاً ، بحيث تكون مجتمعة مايعرف بالأجاكس (AJAX) والتقنيات هي(2):

- HTML,CSS,DHML
- JavaScript
- XML
- XmlHttpRequest

1.1.3 دواعي إستخدام تقنية الاجاكس (AJAX)

لقد أدخلت تقنية الأجاكس تطوراً كبيراً في مجال تطبيقات الويب حيث ساعدت في حل المشاكل المتعلقة بالويب ، كما أنها ساهمت في حل أوجه القصور الموجودة في تقنية التخزين المؤقت (Cache) فلذلك يمكن تلخيص اسباب إستخدامها في الآتي (3) :

- ١ . لاحتياج إلي إعدادات خاصة في المخدم .
- ٢ . تخفيف العبء علي المخدم .
- ٣ . سهولة التعامل مع المخدم ، وذلك بأن يتقاسم معه العميل (client) بعض المعاملات.
- ٤ . خفة التصفح بجعل المتصفح لا يعيد تحميل الصفحات من جديد.
- ٥ . تخفيف العبء علي قواعد البيانات حيث لا يتم استدعاء البيانات الموجودة مرة أخرى.
- ٦ . إمكانية إجراء عملية الإرتباط بين العميل (Client) و المخدم دون أن يكون للمستخدم الإحساس بعملية الإرتباط التي تمت (اي من وراء المستخدم).
- ٧ . إمكانية تحديث الصفحة دون إعادة تحميلها من جديد ، كما يتم إرسال الطلبات التي تحتاج الي تحديث فقط للمخدم.

رغم الإيجابيات السابقة إلا انه تقنية الأجاكس بها السلبيات التالية :

- بالضرورة يجب أن يكون المتصفح يدعم java script ومع هذا يجب أن يدعم

الكائن HTTP Requests.

- ملفات الجافا اسكربت (Java Script) التي يتم تحميلها مع صفحات التطبيق دون ان يعلم بها المستخدم يمكن ان تستغل بصورة سيئة .
- بمأن الصفحة لايعاد تحميلها من جديد فيتم إلغاء الأداة Back و الأداة Forward
- يتم إخفاء عنوان الصفحة التي ياتي منها الرد و ذلك لان هذه الصفحة لاتتم مناداتها مباشرة بل عن طريق برنامج الجافا اسكربت (JavaScript) يتم مناداة الجزء من البرنامج الذي يقوم بتحديد الرد (Response) .

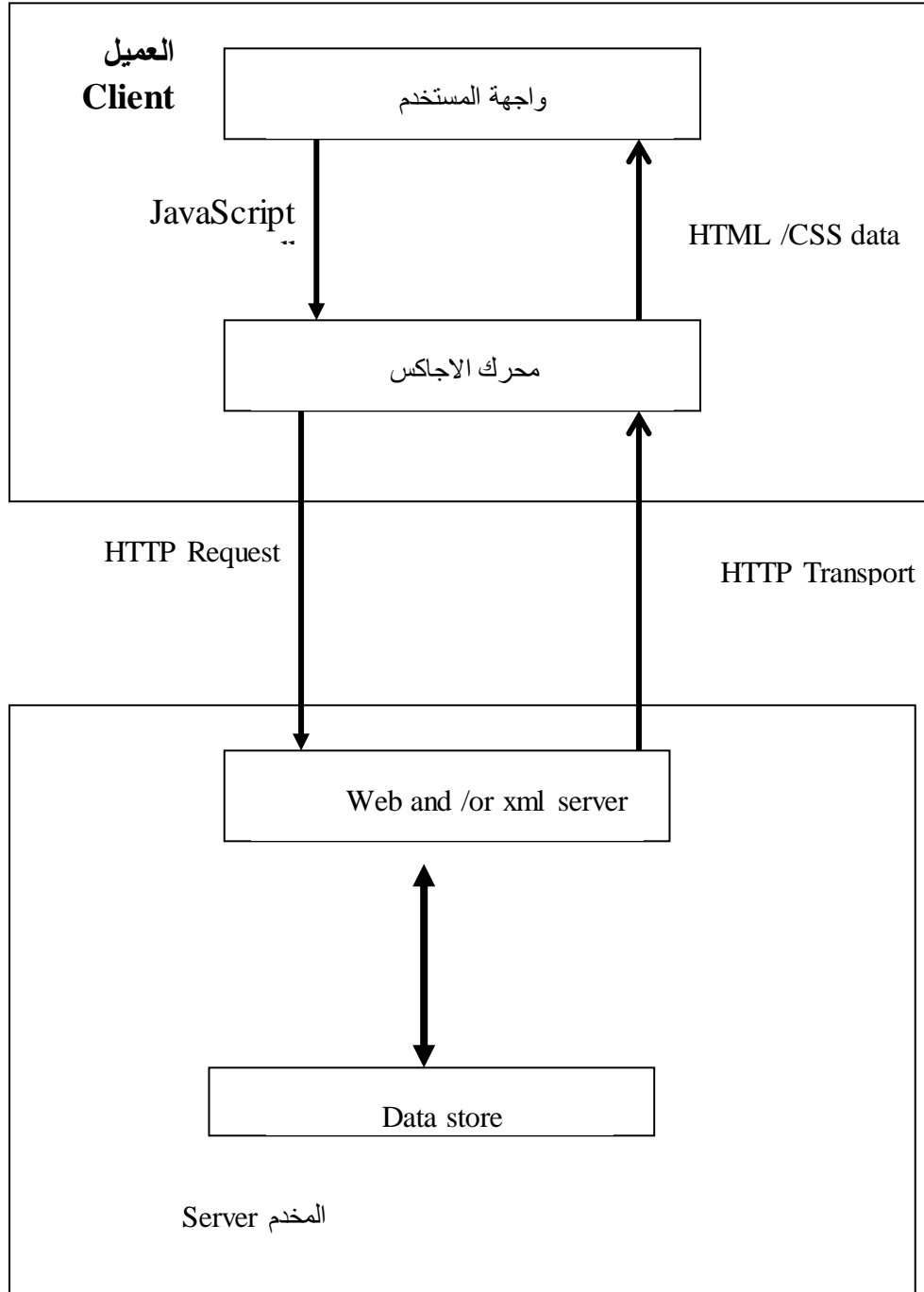
2.1.3 الطريقة العامة لعمل تطبيقات الويب القائمة على تقنية الأجاكس

- في المتصفح تتم كتابة برامج JavaScript لجلب البيانات من المخدم .
- إذا كانت البيانات المراد جلبها من المخدم كبيرة فإنه يتم إستخدام الكائن XMLHttpRequest لإرسال الطلب للمخدم دون أن يتسبب ذلك في تحميل الصفحة الحالية ويمكن للمستخدم ان يواصل بقية عمله في هذه الصفحة .
- البيانات التي تأتي من المخدم يمكن أن تكون ملفات XML أو ملفات نص فقط ويمكن لبرامج JavaScript أن تقرأ هذه البيانات وتضعها للإستخدام .

3.1.3 مقارنة طريقة عمل نموذج تطبيقات الأجاكس و تطبيقات الويب العادية

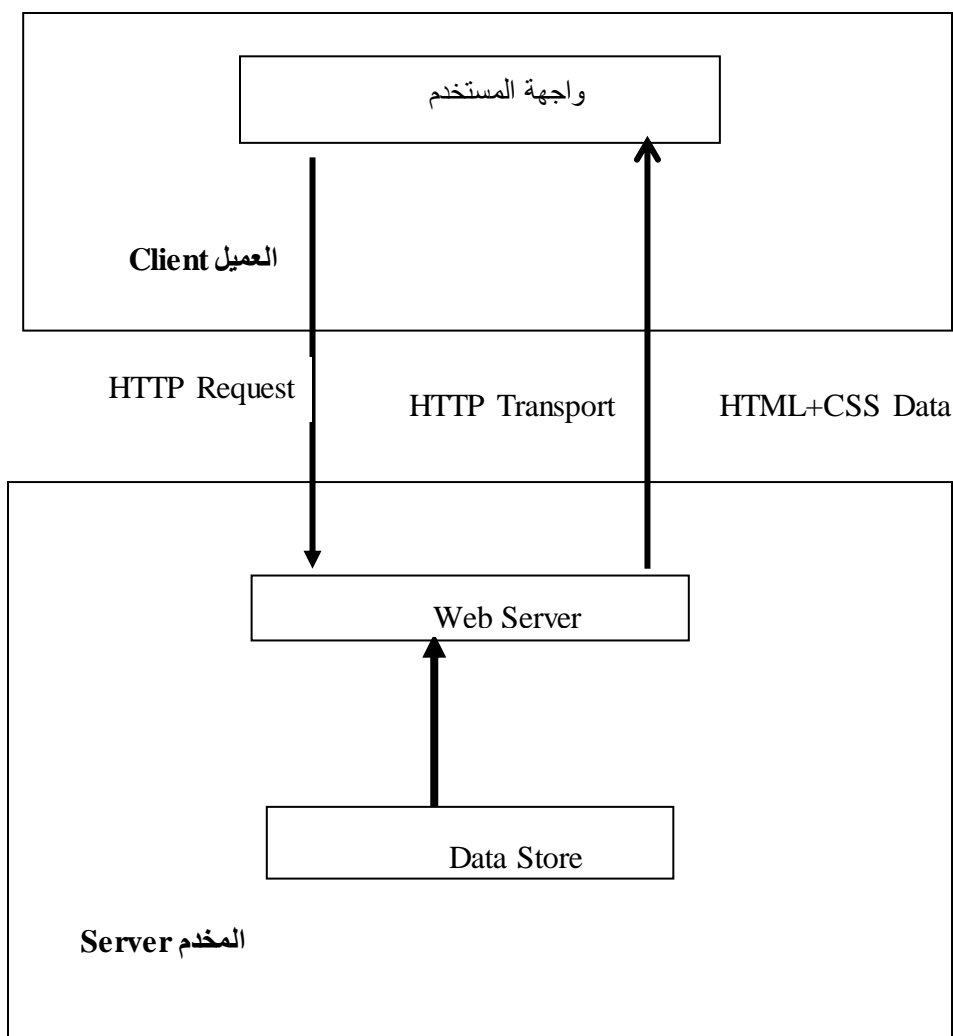
في تطبيقات الويب العادية تكون أغلب نشاطات المستخدم متمثلة في إرسال طلبات إلي المخدم ثم يقوم المخدم بمعالجة الطلبات و الحصول علي البيانات ثم إرسالها في صورة ملفات لغة النصوص التشعبية (HTML) إلي متصفح المستخدم وبالتالي تحديث كامل للصفحة الموجودة في المتصفح وباستخدام هذه الطريقة يكون المستخدم في حالة انتظار حتى يأتيه الرد من المخدم بينما في التطبيقات القائمة علي تقنية الأجاكس فإن الأمر يختلف تماماً عما سبق ، حيث أن الطلبات التي يتم إرسالها إلي المخدم هي الطلبات التي تحتاج إلي تحديث فقط ويتم ذلك بواسطة محرك الاجاكس (AJAX Engine) وهو عبارة عن برنامج JavaScript يتم تحميله مع الصفحة إلي المتصفح دون أن يكون لدى المستخدم علم بعملية التحميل ووظيفة هذا البرنامج هي إرسال الطلبات الى المخدم لحظة كتابتها حيث يتم كتابة أسماء المتغيرات التي سوف يتم إرسالها في هذا البرنامج وبالتالي لاتوجد حوجة لإعادة تحديث كل الصفحة التي يستخدمها المستخدم حالياً

حيث يمكن للمستخدم مواصلة عمله علي ذات الصفحة ، بينما يتم إرسال الطلب وتحديث جزء محدد من الصفحة ، حيث يتم تسمية هذا الجزء ويتولى برنامج الجافا إسكريبت (JavaScript) تحديث هذا الجزء من الصفحة. والشكل (1.3) يوضح نموذج تطبيقات الويب القائمة علي الـ Ajax .



شكل (1.3) نموذج تطبيقات الويب القائمة علي تقنية الـ Ajax.

أما في تطبيقات الويب العادية لا يحدث هذا السيناريو حيث تكون عملية الارتباط بالمخدم مباشرة كما في الشكل (2.3) .



شكل (2.3) نموذج تطبيقات الويب العادية.

4.1.3 التزامن في نموذج تطبيقات الأجاكس و نموذج تطبيقات الويب العادية

في نموذج الأجاكس في كل عملية يقوم بها المستخدم عادة ما يتم توليد طلب HTTP (HTTP Request) و الذي يقوم بالارتباط بمحرك الأجاكس بواسطة برنامج (JavaScript) ولكن ليس كل عملية رد (Response) تحتاج للرجوع للمخدم مثال لذلك أن تتم عمليات كتابة البيانات في الذاكرة وفي عمليات البحث التي يقوم بها المحرك بنفسه فإذا كانت هنالك حاجة لبعض الأشياء في المخدم أو عمليات تقديم بيانات للمعالجة أو عمليات تحميل واجهات استخدام أو بيانات

جديدة فإن المحرك (Ajax Engine) يجعلها متزامنة ويتم ذلك عادة بإستخدام ملفات Xml ولا يوجد عملية إيقاف لعملية تواصل المستخدم مع التطبيق ، إذ يمكن للمستخدم أداء عدة مهام في آن واحد و يؤدي إلي زيادة الأداء بالنسبة للمستخدم وكذلك تقل كمية البيانات المنقولة عبر الشبكة .

2.3 الأدوات المستخدمة في التصميم بإستخدام تقنية الأجاكس

كما ذكر سابقاً فإن تقنية الأجاكس عبارة عن مجموعة من التقنيات وكل من هذه التقنيات يؤدي وظيفة معينة و التقنيات هي :

1.2.3 لغة JavaScript

تستخدم الآن في ملايين صفحات الويب لعدة أغراض منها :

- تطوير التصميم
- عمليات التحقق في النماذج (Form).
- إكتشاف المتصفحات.

كما أنها تؤدي عدة أغراض أخرى فلذلك هي من اللغات المفضلة في تطبيقات الإنترنت ولذلك فإنها تستخدم في معظم المتصفحات مثل Internet Explorer ، ، Mozilla وكذلك في عدد من المتصفحات الأخرى .

1.1.2.3 الخدمات و المزايا التي تقدمها الجافا إسكربت (JavaScript)

- تعطي المصممين الذين يستخدمون لغة النصوص التشعبية (HTML) أداة لغة برمجية وذلك لأن (HTML) بطبيعتها لغة ترميزية وليست برمجية.

- تمكن من وضع نصوص متغيرة في صفحات HTML ومثال لذلك
`Document.write("<h1>+name+"</h1>")` حيث يمكن بإستخدام هذه الجملة طباعة
 قيمة المتغير `name`.
- توفر إمكانية التعامل مع الأحداث التي تنشأ نتيجة لإستخدام التطبيق فيمكنها أن تجعل
 التطبيق يتغير وفقاً لنشاطات المستخدم .
- يمكن إستخدام JavaScript لقراءة و كتابة مكونات (HTML).
- تستخدم في عمليات التحقق من القيم المرسلة للمخدم قبل إجراء عملية إرسالها.
- إذا كان حجم البيانات المراد إسترجاعها من المخدم متعددة فإنها (JavaScript) تسمح
 للكائن XMLHttpRequest بالإرتباط بالمخدم دون أن يكون لدى المستخدم الإحساس
 بعملية الإرتباط إضافة إلي ذلك فإن هذه العملية تتم دون أن يتم إيقاف نشاط المستخدم
 في إستخدام التطبيق.

2.1.2.3 وضع عبارات JavaScript داخل ملف HTML

يمكن لملف HTML أن يحتوي علي العبارات البرمجية المكتوبة بلغة JavaScript والشكل (3.3)
 يوضح وضع عبارة بسيطة مكتوبة داخل ملف HTML باستخدام JavaScript .

```
<html >
<body >
<Script type="text/javascript">
Document.write("hello javascript")
</script>
</body>
</html>
```

شكل (3.3) وضع عبارات JavaScript داخل ملف HTML

2.2.3 لغة XML (eXtensible Markup Language)

هي إحدى اللغات الوصفية للبيانات وتتميز عن HTML في الآتي (1) :

- XML صممت لتختص بوصف البيانات بينما HTML تختص بكيفية عرض البيانات.
- عدد الأوسام (Tags) في HTML محدودة و معروفة ولكل واحد وظيفة معينة يؤديها
- أما في XML فعددها غير محدد فكل مستخدم يستطيع ان يعرف ما يحتاج اليه من أوسام (tags) لوصف بياناته.
- تتيح XML إمكانية التحقق من نوعية البيانات ويتم ذلك بإستخدام DTD و XML Schema.
- يمكن عرض البيانات التي توجد في الأوسام (tags) في صورة HTML بإستخدام Xslt.
- XML مكمل HTML وليست بديلة لها.
- بإستخدام XML يمكن الفصل بين طريقة عرض البيانات ومحتوي البيانات.
- تستخدم XML لتبادل البيانات حيث يمكن إستخدامها في الأنظمة غير المتوافقة مع بعضها البعض .
- يمكن إستخدام XML في البيانات المشتركة حيث ان ملف البيانات يمكن إستخدامه بواسطة عدد من المستخدمين.
- بإستخدام XML تكون البيانات مفيدة أكثر حيث أنها تكون متاحة لعدد من المستخدمين.
- من المصادر المفتوحة (open source).

1.2.2.3 قواعد بناء عناصر XML

- يمكن أن يحتوي الإسم علي حروف ،أرقام ورموز أخرى .
- الإسم لا يحتوي علي مسافات فارغة.
- يجب أن يوضع بين العلامتين <name_of tag>.
- تعتبر حساسة لحالة الأحرف حيث تفرق بين الحروف الكبيرة و الصغيرة.
- كل عنصر يجب أن يكون له نهاية على الشكل </name_of tag>

2.2.2.3 الشكل العام لملف XML

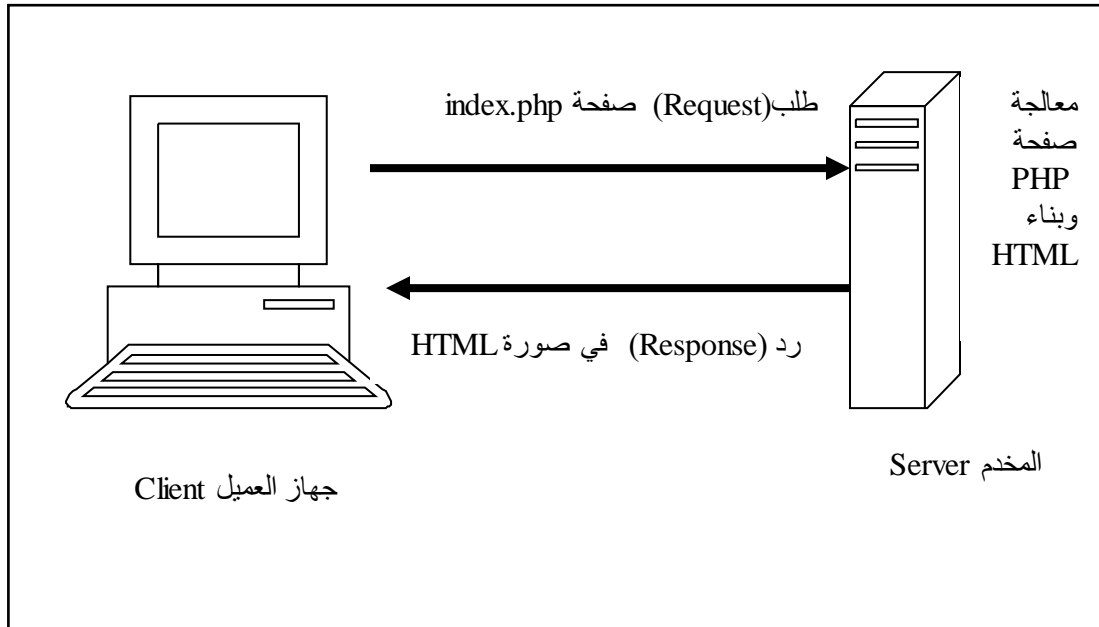
يتكون من مجموعة من الأوسام (Tags) حيث لكل وسم (Tag) رمز بداية و نهاية وبينهما البيانات المراد وصفها . يمكن وصف ذلك في الشكل (4.3)

```
<root_element>
<child_element >data.....</child_element>
</root_element>
```

شكل (4.3) الشكل العام لملف XML.

3.2.3 لغة PHP (Personal Home Page)

إحدى المصادر المفتوحة (open source) والتي تختص في برمجة العمليات التي تحدث في المخدم ،حيث يمكن بواسطتها إجراء كل العمليات علي البيانات في قواعد البيانات من إدخال،تحديث،إستعلام ، حذف. فعلى سبيل المثال إذا كانت صفحة تسمى index.php تخص تطبيق معين فعند إجراء عملية طلب لهذه الصفحة كما في الشكل (5.3) في المتصفح ، فبعد أن يقوم المخدم بمعالجة البيانات المطلوبة في الصفحة يجب أن يعيد الرد (Response) للمستخدم في صورة HTML أو بأي لغة يفهمها جهاز العميل (client) .



شكل (5.3) عملية الطلب (Request) والرد (Response) باستخدام PHP.

1.3.2.3 دواعي استخدام PHP

بما أن لغات برمجة تطبيقات الويب كثيرة العدد و التي يمكن أن تؤدي وظائف (PHP) فيتم

إستخدام (PHP) للأسباب الآتية (4) :

- من المصادر المفتوحة
- بعد أن أصبح أغلب مستخدمي تطبيقات الويب يسعون إلي تقليل تكاليف تسكين برامجهم ،إتجه مصممي تطبيقات الويب إلي المصادر المفتوحة والتي يقوم بتطويرها مطوري الويب حيث لا يوجد عملية دفع عند إستخدام إحدى أدوات المصادر المفتوحة.
- قابلية التوسعة
- يمكن توسعة مترجم PHP بسهولة وإضافة إليه ميزات أخرى وذلك باستخدام لغة C وذلك لان الشفرة التي كتب بها مفتوحة.
- سريعة وسهلة
- إن عملية المقارنة بين اللغات من حيث السرعة عملية صعبة فعلي سبيل المثال إن البرامج المكتوبة بلغة C تكون سريعة ولكنها صعبة لحد ما ، كما أنها طويلة مقارنة باللغات الأخرى فلغة PHP سريعة مقارنة باللغات الأخرى كما أن سياقها قريب جداً من لغات ASP،JSP.
- متعددة بيئات العمل
- عند إجراء عملية تنفيذ لبرنامج PHP فيمكن تنفيذه في عدة بيئات مختلفة منها :
 - Windows 2000/NT
 - Unix.
 - مع كل من IIS، Apache وفي عدد من المتصفحات.

Apache 4.2.3

يعتبر من مخدمات الويب (Web Server) المحببة لدى أغلب مصممي تطبيقات الويب إذ أنه من المصادر المفتوحة إضافة إلى ذلك أنه لا يعمل فقط تحت بيئة Unix فيمكن له أن يعمل في بيئة Windows إضافة إلى ذلك انه سريع .

(Document Object Model) DOM 5.2.3

عبارة عن تمثيل كائني موجه لملفات XML و HTML حيث تقوم بتمثيل الهيكل العام لملف XML و HTML لكائنات متدرجة لإجراء عملية إعراب لها بواسطة أدوات XML القياسية(1)

MYSQL 6.2.3

إحدى المصادر المفتوحة وتخص أنظمة قواعد البيانات كما تتميز بعدة مزايا من بينها :

- سهولة الإستخدام.
- يمكن إستخدامها مع أدوات تجعلها عملية.
- من المصادر المفتوحة (open source).
- تعمل في مختلف البيئات .
- لا تحتاج إلى ذاكرة كبيرة.
- سريعة في عمليات الإسترجاع وغيرها من العمليات التي تجرى على البيانات.
- يمكن أن تستخدم عندما يكون حجم البيانات المراد تخزينها كبير .

XmlHttpRequest Object 7.2.3

يعتبر العمود الفقري في التطبيقات القائمة على تقنية الأجاكس (AJAX) ، فيجب قبل إرسال طلب (Request) أو إستقبال رد (Response) من المخدم ، إستخدام JavaScript لإنشاء كائن XmlHttpRequest فهو صيغة غير قياسية من W3C فعند إستخدام Internet Explorer كمتصفح فيتم تعريفه من نوع ActiveX أما المتصفحات الأخرى مثل سفاري Safari فيتم تنفيذه ككائن من نوع JavaScript ونسبة لهذه الاختلافات فيجب أن تحتوي JavaScript على منطق لإنشاء XmlHttpRequest بإستخدام تقنية ActiveX أو إستخدام كائن JavaScript (2).

١,٧,٢,٣ كيفية إنشاء الكائن XmlHttpRequest

```
Var xmlhttp;  
Function createXMLHttpRequest()  
{  
If(window.ActiveXObject)  
{  
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");  
else if(window.XMLHttpRequest)  
{xmlhttp=new XMLHttpRequest();  
}  
}
```

شكل (6.3) كيفية إنشاء كائن XMLHttpRequest (2)

يلاحظ من العبارات السابقة إن عملية إنشاء XmlHttpRequest تتطلب أولاً تعريف متغير عام يدعى xmlhttp وذلك لتخزين مرجع لهذا الكائن. أما الدالة المسماة createXMLHttpRequest تقوم بالفعل بإنشاء هذا الكائن ، فالدالة تحتوي علي عبارات منطقية بسيطة وذلك لتحديد كيفية إنشاء هذا الكائن ، فعملية الإتصال window.ActiveXObject ترجع هذا الكائن أو لاشي (Null) ويحدد ذلك بواسطة عبارة if فإذا كان المتصفح يدعم ActiveX فهو Internet Explorer والكائن

XMLHttpRequest سوف يتم إنشائه حيث يتم ذلك بتمرير نص يوضح نوع الكائن XMLHttpRequest. أما إذا فشلت العبارة window.ActiveXObject فان عبارة عدا ذلك (else) تنفذ. فتتم معرفة أن المتصفح يدعم XMLHttpRequest ككائن من نوع JavaScript.

2.7.2.3 الدوال الموجودة في الكائن XMLHttpRequest

يوجد عدد من الدوال في الكائن XMLHttpRequest ولكل منها وظيفة تؤديها. الجدول (1.3) يوضح وصف هذه الدوال (2) .

الوصف	الدالة
إيقاف الطلب الحالي	Abort()
ترجع نص هو قيمة الترويسة الرئيسية المحددة	getResponseHeader("header")
ترجع كل الترويسة الرئيسية لنتيجة طلب HTTP	getAllResponseHeader()
تقوم بوضع مرحلة الارتباط بالمخدم حيث يمكن للدالة أن تكون get،post أو put أما url فيمكن أن يكون أي عنوان يتم تحديده	Open("method","url")
إرسال الطلب للمخدم	Send(content)
تقوم بوضع القيمة المعطاة للعنوان المحدد	setRequestHeader("header","value")

جدول (1.3) الدوال الموجودة في الكائن XMLHttpRequest (2)

3.7.2.3 الخصائص القياسية في الكائن XMLHttpRequest

يوجد عدد من الخصائص القياسية في كائن XmlHttpRequest والتي تعبر عن الحالة الحالية فيمكن وصفها في الجدول (2.3)

الخاصية	الوصف
Onreadystatechange	يتم معالجة الحدث في كل عملية تغيير للحالة وكأنها عملية مناداة لدالة javascript
Readystate	تأخذ القيم (٠ لحالة البداية، ١ يتم التحميل، ٢ تمت عملية التحميل، ٣ تعمل الآن، ٤ اكتملت)
responseText	تحدد إن الرد الذي سوف يأتي من المخدم عبارة عن نص
responseXML	تحدد إن الرد عبارة عن XML
Status	تعبّر عن حالات HTTP حيث ٢٠٠ لعملية OK، ٤٠٤، ٤٠٤ لعملية NotFound
statusText	لتحدد إصدار النص في HTTP

جدول (2.3) الخصائص القياسية في الكائن XmlHttpRequest (1)

4.7.2.3 إجراءات إرسال كائن XmlHttpRequest

تتضمن عملية إجراءات إرسال كائن XMLHttpRequest الآتي (1)

- تضمين دالة JavaScript في أحداث HTML.
- إنشاء كائن XMLHttpRequest في دالة JavaScript.
- فتح كائن XMLHttpRequest لتحديد العنوان المحدد ودالة HTTP.
- تسجيل عملية حدوث رجوع وذلك عند إكمال الطلب.
- إرسال كائن XMLHttpRequest متزامناً .
- إسترجاع الرد (Response) كهيئة Xml ثم تغيير صفحة HTML.

3.3 كيفية معالجة الرد الذي يأتي من المخدم (Processing the Server Response)

يوفر كائن XMLHttpRequest خاصيتين للتعامل مع الرد الذي يأتي من المخدم هما :

- الرد النصي :

في هذه الحالة يتم التعامل مع الرد كنص عادي.

- التعامل مع الرد كهيئة XML.

فعندما تكون حالة الرد بسيطة فالأجدي التعامل معها كنص بسيط ومثال لذلك عندما نستخدم

مربع التحذير (Alert Box) لعرض رسالة تحذير كرد قادم من المخدم (2) .

1.3.3 إستخدام خاصية innerHTML لإنشاء محتوى ديناميكي

إن عملية معالجة الرد الذي يأتي من المخدم كنص عادي ليس بالمرونة الكبيرة حيث أن طريقة

النص تعاني من عدة أمور منها البنية التركيبية كما توجد صعوبة في عملية إعرابه (Parsing)

بإستخدام JavaScript كما أن عملية إنشاء صفحة متغيرة المحتوى في غاية الصعوبة. يمكن

إعتبار طريقة النص العادي للتعامل مع الرد كفوءة إذا تم إستخدامها مع خاصية innerHTML

لعناصر HTML حيث يتم إستخدام نص بسيط لتمثيل المحتوى بين مجموعة رموزالبداية

والنهاية وذلك بإستخدام النص و innerHTML يمكن للمخدم إنتاج محتوى

HTML يعالج في المتصفح باستخدام خاصية innerHTML (2)

4.3 التعامل مع الأحداث التي تنشأ في المتصفح

إن طبيعة التطبيقات المتفاعلة تحتم على برامجها المكتوبة التفاعل مع الأحداث التي تحدث في

المتصفح نتيجة لإستخدام التطبيق ومثال لتلك الأحداث، الضغط على أيقونة ،رابط ، تحريك

الماوس أو تحميل صفحة . الجدول (3.3) يوضح بعض هذه الأحداث ودواعي ظهورها.

الحدث	الوصف
Onabort	يحدث عندما يرفض المستخدم حدثاً معيناً
Onchange	يحدث عندما تتغير البيانات في أداة تحكم معينة مثل مربع نص
Onclick	يظهر عند الضغط على عنصر معين
Onerror	يحدث عندما يكون هنالك خطأ في عبارات JavaScript
Onload	يحدث عند تحميل الصفحة
Onsubmit	نتيجة للضغط على أيقونة الإرسال يتولد هذا الحدث
Onunload	يحدث عندما لا يحمل المتصفح الصفحة وينتقل لصفحة أخرى
Onkeypress	يظهر عند الضغط على أي مفتاح
Onmouseout	يظهر عندما يحرك المستخدم زر الماوس عند عنصر محدد
Onmousedown	يظهر هذا الحدث عند تحريك مؤشر الماوس أسفل أيقونة معينة

جدول (3.3) بعض الأحداث التي تنشأ في المتصفح عند إستخدام تطبيق معين.

5.3 ملخص الفصل

في هذا الفصل تمت مناقشة مفهوم تقنية الأجاكس (AJAX) (HTML,CSS,DHML,JavaScriptXMLHttpRequest) و الأدوات المكونة لها والدور الذي تلعبه كل أداة إضافة لذلك تم توضيح التزامن في كل من تطبيقات الويب القائمة على الطريقة العادية لعمل الويب و التطبيقات القائمة على تقنية الأجاكس، كما تم شرح بعض الأحداث التي تنشأ في المتصفح ودواعي ظهور كل حدث.